



**VK202-25**  
**Technical Manual**

**Revision: 3.0**

# Contents

<b>Contents</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What to Expect From the VK202-25 . . . . .	1
1.2 What Not to Expect From the VK202-25 . . . . .	1
1.3 Keypad Interface . . . . .	1
1.4 Setup for Testing . . . . .	2
1.5 Trying Out the VK202-25 . . . . .	3
1.6 Trying Out a Keypad . . . . .	3
1.6.1 Here's what to Do . . . . .	3
1.7 Manual Over-ride . . . . .	4
<b>2 Connections</b>	<b>4</b>
2.1 Connector Pinout . . . . .	4
2.1.1 Power Connections . . . . .	6
2.1.2 Five Volt Modules . . . . .	6
2.1.3 Wide Voltage Range Modules . . . . .	7
2.2 Communication Connection . . . . .	8
2.2.1 RS-232 Communications . . . . .	8
2.2.2 TTL Communications . . . . .	10
2.2.3 I <sup>2</sup> C Communications . . . . .	12
2.2.4 ACK . . . . .	13
2.2.5 Dallas 1-Wire Bridge . . . . .	13
2.3 General Purpose Outputs . . . . .	14
<b>3 Displaying Text</b>	<b>15</b>
3.1 General . . . . .	16
3.2 The Built In Character Font . . . . .	16
3.3 Writing Text to the Display . . . . .	17
3.4 Text Commands . . . . .	17
3.4.1 Auto line wrap on (254 67) (R) . . . . .	17
3.4.2 Auto line wrap off (254 68) (R) . . . . .	17
3.4.3 Auto scroll on (254 81) (R) . . . . .	17
3.4.4 Auto scroll off (254 82) (R) . . . . .	17
3.4.5 Set cursor position (254 71 [column] [row]) . . . . .	18
3.4.6 Send cursor home (254 72) . . . . .	18
3.4.7 Turn on underline cursor (254 74) (R) . . . . .	18
3.4.8 Turn off underline cursor (254 75) (R) . . . . .	18
3.4.9 Turn on block (blinking) cursor (254 83) (R) . . . . .	18
3.4.10 Turn off block (blinking) cursor (254 84) (R) . . . . .	18
3.4.11 Cursor left (254 76) . . . . .	18
3.4.12 Cursor right (254 77) . . . . .	18

<b>4</b>	<b>Keypad Interface</b>	<b>19</b>
4.1	General . . . . .	19
4.2	Connections . . . . .	19
4.3	I <sup>2</sup> C Interface . . . . .	20
4.4	RS-232 Interface . . . . .	20
4.5	Keypad Commands . . . . .	20
4.5.1	Auto repeat mode on (254 126 [mode]) (R) . . . . .	21
4.5.2	Auto repeat mode off (254 96) (R) . . . . .	21
4.5.3	Auto transmit keypresses on (254 65) (R) . . . . .	21
4.5.4	Auto transmit keypresses off (254 79) (R) . . . . .	22
4.5.5	Clear key buffer (254 69) . . . . .	22
4.5.6	Poll keypad (254 38) . . . . .	22
4.5.7	Set debounce time (254 85 [time]) (R) . . . . .	22
<b>5</b>	<b>Bar Graphs and Special Characters</b>	<b>22</b>
5.1	Command List . . . . .	23
5.1.1	Initialize wide vertical bar graph (254 118) . . . . .	23
5.1.2	Initialize narrow vertical bar graph (254 115) . . . . .	23
5.1.3	Draw vertical bar graph (254 61 [column] [height]) . . . . .	23
5.1.4	Initialize horizontal bar graph (254 104) . . . . .	23
5.1.5	Draw horizontal bar graph (254 124 [column] [row] [dir] [length]) . . . . .	23
5.1.6	Define custom character (254 78 [c] [8 bytes]) . . . . .	24
5.1.7	Initialize Medium Digits (254 109) . . . . .	25
5.1.8	Draw Medium Digits (254 111 [row][column][digit]) . . . . .	25
<b>6</b>	<b>Miscellaneous Commands</b>	<b>25</b>
6.1	Command List . . . . .	25
6.1.1	Remember (254 147 [01]) . . . . .	25
6.1.2	Clear display (254 88) . . . . .	25
6.1.3	Set brightness (254 89 [brightness]) (R) . . . . .	26
6.1.4	Set brightness and Save (254 145 [brightness]) (R) . . . . .	26
6.1.5	Display on (254 66 [minutes]) (R) . . . . .	26
6.1.6	Display off (254 70) (R) . . . . .	26
6.1.7	Load startup screen (254 64 [40 characters]) . . . . .	26
6.1.8	General purpose output off (254 86 [gpo #]) . . . . .	27
6.1.9	General purpose output on (254 87 [gpo #]) . . . . .	27
6.1.10	Remember GPO State (254 195 [gpo#][gpo value]) . . . . .	27
6.1.11	Set I <sup>2</sup> C address (254 51 [address]) (R) . . . . .	27
6.1.12	Read module type (254 55) . . . . .	28
6.1.13	Set RS-232 port speed (254 57 [speed]) (R) . . . . .	28
6.1.14	Set Serial Number (254 52 [byte1] [byte2]) . . . . .	28
6.1.15	Read Serial Number (254 53) . . . . .	29
6.1.16	Read Version Number (254 54) . . . . .	29
6.2	Flow Control . . . . .	29
6.2.1	Enter Flow Control Mode (254 58 [full][empty]) . . . . .	29
6.2.2	Exit Flow Control Mode (254 59) . . . . .	29

<b>7</b>	<b>1-Wire Commands</b>	<b>30</b>
7.1	Device Identification . . . . .	30
7.2	Protocol . . . . .	30
7.3	ROM Commands . . . . .	30
7.4	Display 1-Wire Connections . . . . .	31
7.4.1	Transaction command (254 C8 1 [flags] [Send Bits] [Recieve bits] [Send data]) . . .	31
7.4.2	Search command (254 200 2) . . . . .	33
<b>8</b>	<b>Appendix: Command Summary</b>	<b>33</b>
8.1	General . . . . .	33
8.2	Issuing Commands . . . . .	33
8.3	On Numbers . . . . .	34
8.3.1	ASCII Characters . . . . .	34
8.4	Text Commands . . . . .	35
8.5	Keypad Interface Commands . . . . .	37
8.6	Bar Graphs and Special Characters . . . . .	38
8.7	Miscellaneous Commands . . . . .	39
<b>9</b>	<b>Appendix: Specifications and Options</b>	<b>42</b>
9.1	Specifications . . . . .	42
9.2	Options . . . . .	43
<b>10</b>	<b>Appendix: Glossary</b>	<b>43</b>

# 1 Introduction

The VK202-25 comes equipped with the following features:

- 20 column by 2 line text vacuum fluorescent display
- Built in font with provision for up to 8 user defined characters
- Speeds from 1200 bps to a lightning fast 19.2 Kbps over RS-232
- Communication over RS-232 or I<sup>2</sup>C
- Use of up to 127 modules on the same 2 wire I<sup>2</sup>C interface
- 1-wire bus that is capable of communicating with many devices over a single wire plus a ground reference
- Fully buffered so that no delays in transmission are ever necessary
- Ability to add a customized splash / startup screen
- Software controlled brightness with configurable time-out setting up to 90 minutes and software controlled speed
- Use of up to a 25 key keypad with a 10 key buffer
- Six general purpose outputs for a variety of applications
- Horizontal or vertical bar graphs
- Variable power options, +5V or +9V to +15V
- Extended temperature option
- Fits Matrix Orbital's PC Bay inserts without any modifications

## 1.1 What to Expect From the VK202-25

The display is designed as the display unit for an associated controller. The controller may be anything from a single board, special purpose micro-controller to a PC, depending on the application. This controller is responsible for what is displayed on the screen of the display.

The display provides a simple command structure to allow text and bar graphs to be displayed on the screen. Text fonts are built in, and use standard ASCII mapping. Provision is made for up to 8 user defined characters.

The screen is luminous for low light situations. Display may be turned on or off under program control. Brightness is adjustable to compensate for differing lighting conditions.

General purpose outputs allow the controller to switch up to six electronic or electro-mechanical devices by issuing commands to the display unit. These can be used for controlling LEDs, relays, etc.

## 1.2 What Not to Expect From the VK202-25

The display does not include bitmap graphics capability, except that permitted by defining special characters.

## 1.3 Keypad Interface

The keypad interface takes row / column input and converts it to ASCII characters, which are delivered out the RS-232 or I<sup>2</sup>C port to the associated controller. Note that the keypad is not used to directly control

any aspect of the operation of the display, which acts simply as a matrix to serial converter. To use the keypad to control the display, the controller must be programmed accordingly.

## 1.4 Setup for Testing

Before setting up the application the user may want to try out the display. This is easily done with a PC. If not equipped with a dual bay PC mounting kit, the following will be required;

- A 4-pin power connector of the type used to connect 3.5" floppy drive. Take care not to connect the display to an unmodified spare power connector in a PC.
- A 5V power supply.
- A PC with a spare RS-232 port (COM1 or COM2).
- A 9 or 25 pin RS-232 serial cable. If using a 25 conductor cable, a 9 to 25 pin adapter will be required.

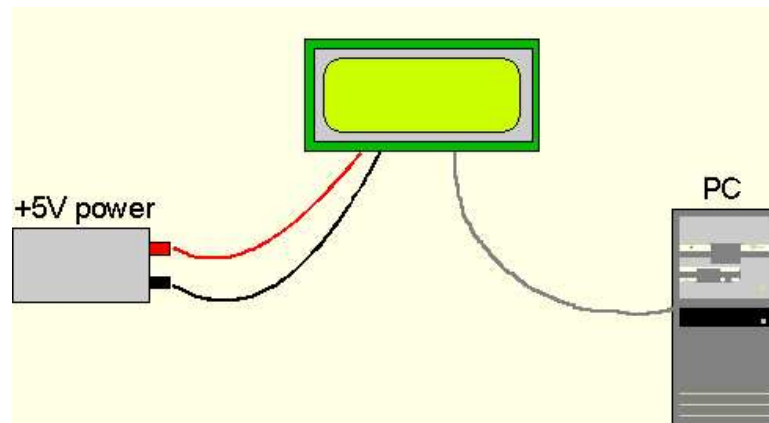


Figure 1: Connections for Testing

1. Refer to the Figure above for the following steps.
2. Wire the connector to the power supply. On most connectors the RED lead will go to +5V and the BLACK lead to GND.

---

**NOTE** The Manufacturer's Warranty becomes void if the unit is subjected to over-voltage or reversed polarity.

---

3. Connect the display to the PC using the serial cable and adapter if required. Make sure the RS-232 cable includes the required ground lead. There must be no voltage differential between the RS-232 ground and the power supply ground.
4. Connect the power connector, making sure that the +5V goes to V+. Turn on the power: the display should come on.

## 1.5 Trying Out the VK202-25

The unit should be connected to power. The PC and display should be on. To experiment with typing text, run a PC terminal program, such as Hyperterm. Make sure it's configured to use the correct port. Set the baud rate to 19,200. If characters are typed on the keyboard, they should now appear on the display screen. Text will wrap around to the next line when the end of a line has been reached.

A few common ASCII control characters work as follows:

Table 1: ASCII Control Characters

Character	Hex value	Function
CR	0x0D	Moves cursor to beginning of the current line.
LF	0x0A	Moves cursor to beginning of the next ( or previous ) line.
FF	0x0C	Clears the display and puts the cursor at the top left
BS	0x08	Moves the cursor one position to the left and clears that position.

---

**NOTE** These command characters are not guaranteed to work on other Matrix Orbital display modules.

---

To exercise some of the other features of the display, a program (in any convenient language such as Basic or C) will need to be written in order to issue the required command strings. Most terminal programs are unable to issue the 0xFE character needed as a command prefix.

## 1.6 Trying Out a Keypad

Since a number of different keypad types can be connected to the display, the results may be a little unpredictable. At this point all we need to do is make sure that the keypad and interface work, and possibly generate an ASCII map for any programming requirements. The keypad interface on the display converts a row / column connection to an ASCII character. By default, a keypress is transmitted as serial data immediately. Keypad buffering can be selected using the appropriate commands.

### 1.6.1 Here's what to Do

1. The PC should be running a terminal program, such as Hyperterm.
2. With the display connected to the PC, plug in the keypad. If the connector has fewer pins than the one on the display, center it as well as possible.

---

## NOTES

- The keypad connector must be wired with columns on one side and rows on the other side of the center of the connector. If the keypad isn't wired this way an adapter must be made or the connector must be rewired to meet this requirement.
  - The connector is reversible. Reversing the connector will not damage the keypad or the display, but it will change the ASCII character map.
- 

3. Press a key on the keypad. An upper case ASCII character (A-Y) should appear on the PC screen. Different keys should generate different characters.

To experiment, reverse the connector and see if it generates a more logical set of characters. Ultimately, the program in your micro-controller will have to 'map' these characters to the ones marked on the keypad, which will likely be different.

## 1.7 Manual Over-ride

Manual over-ride should only be required in one instance. If for some reason the module is set at a baud rate which cannot be produced by the host system and all communication to the display is lost, then the user should follow this simple procedure:

1. Load an appropriate test software that is located on the eCD (ex: displaytuner.exe)
2. Turn off the display.
3. Put a jumper on pins 5 and 6 of the keypad connector (C5 and R1).
4. Power up the display. The baud rate is now set to 19200 baud.
5. Remove the jumper and change the RS232 port settings to the desired baud rate. Refer to the "Set RS-232 port speed" command for acceptable baud rates.
6. Use "Display Tuner" or any other available test software to check functionality of the unit.  
Try turning the display off/on, turning the brightness up/down, etc.
7. Turn off the display.
8. Power up the display.

## 2 Connections

### 2.1 Connector Pinout

Refer to the diagram below for this section.



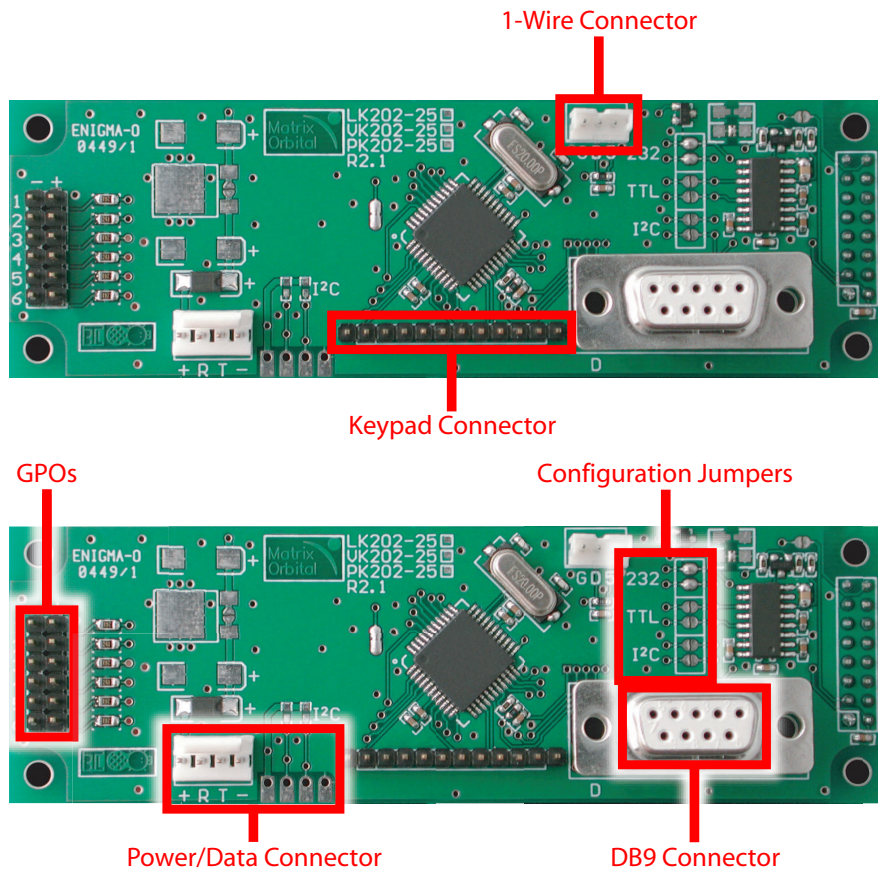


Figure 2: Electrical Connections

The display has four connectors:

Table 2: Connectors & Functions

Connector	Function
12 pin dual header	General purpose outputs (6)
4 pin	Power (5.0 VDC), Data (Rx, Tx), I <sup>2</sup> C communications and Ground
12 pin header	Keypad and ground strapping
DB-9F	RS-232 / power

## 2.1.1 Power Connections

Power is applied via pin1 and ground via pin 4 as shown in the figure below. Power requirement is +5 VDC  $\pm$ 0.25V. As an alternate power connection, power may also be supplied via the RS-232 connector as described in section 2.1.4.

---

### WARNINGS



- Do not apply any power with reversed polarization.
  - Do not apply any voltage other than the specified voltage.
  - Do not use any cables other than the cables supplied by Matrix Orbital, unless aware of the modifications required.
  - Do not apply voltage to the DB-9 connector AND power connector
  - Do not apply more than +5Vdc to pin 9 on the DB-9 connector.
- 

Connector pinout is as follows:

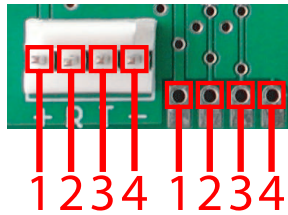


Figure 3: Power Connector

Table 3: Connector Pinout

Pin 1	+5.0 VDC (+7 to +15 VDC with wide voltage option)
Pin 2	SCL (I <sup>2</sup> C clock), Rx
Pin 3	SDA (I <sup>2</sup> C data), Tx
Pin 4	Ground

## 2.1.2 Five Volt Modules

If the display is used in a PC it becomes tempting to plug a spare power connector into the unit. **Don't do this!** Wiring for the PC power connector and that required for the display are different as shown in the Figure below.

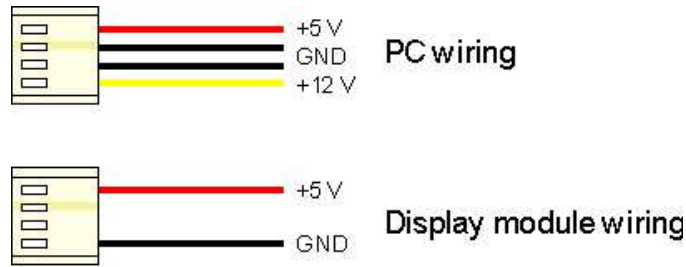


Figure 4: Wiring for 5V Modules

Matrix Orbital can supply an adapter cable designed for use with the display when it's installed in a PC. The cable is wired as shown in the Figure below. **Note that this cable does not provide connections for I<sup>2</sup>C.**



Figure 5: 5V Power Cable

### 2.1.3 Wide Voltage Range Modules

---

**NOTE** Do not use this cable unless the display module has the "Wide voltage range" option (option V). Use of the 12 volt power cable with 5 volt modules will damage the module.

---

The 12 volt power cable is designed for use with wide voltage range display modules mounted in a PC. Wiring required for the 12 volt power connector is shown in the Figure below.

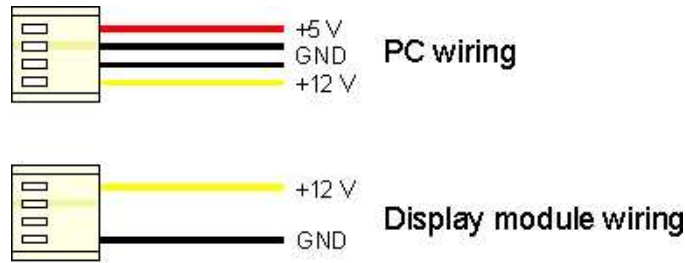


Figure 6: Wiring for 12V Modules

Matrix Orbital can supply an adapter cable designed for use with the display module when it's installed in a PC. The cable is wired as shown in the Figure below.



Figure 7: 12V Power Cable

## 2.2 Communication Connection

### 2.2.1 RS-232 Communications

A standard DB-9F is provided for RS-232 communications. This unit is set to RS232 communication by factory default.

---

#### WARNINGS



- Do not apply any voltage other than the specified voltage.
  - Do not apply voltage to the DB-9 connector AND power connector.
  - Do not apply more than +5VDC to pin #9 on the DB-9 connector if module is designated "standard voltage". If the module is designated "wide or extended voltage", then you may apply between 8-15 Volts accordingly. Please see the figures below for power connections and appropriate solder jump required.
-

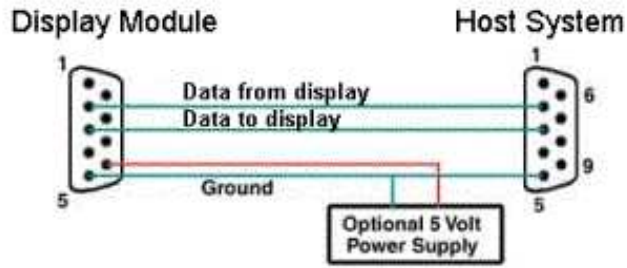


Figure 8: RS-232 and Power Connector

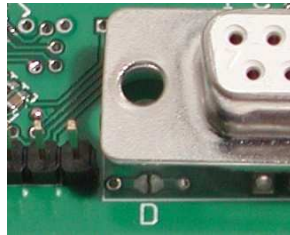


Figure 9: Power Applied via DB9

The RS-232 connector on the PC cable is wired so that a standard ‘straight through’ 9 pin D-sub cable may be used to connect the module to a standard serial port such as COM ports on PCs. Please see the figure below for pinout configuration.

---

**NOTE** This device complies with the EIA232 standard in that it uses signal levels from +/- 3V to +/- 12V. It will not operate correctly at TTL (0 to +5V) levels with out modification. A null modem cable will not work.

---

Table 4: RS-232 Pinout

Pin Number	Direction	Description	LCD	Host
2	Data from LCD	Data out (LCD)	Tx	Rx
3	Data to LCD	Data in (LCD)	Rx	Tx
5	-	Ground	gnd	gnd

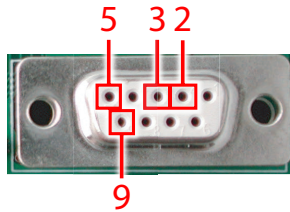


Figure 10: DB9 Pinout

For RS232 communication via the 4 pin connector, please see the figure and table below.

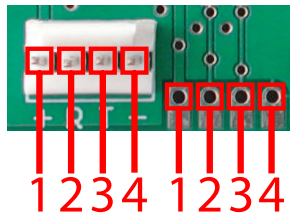


Figure 11: Power Connector

Table 5: Connector Pinout

Pin 1	+5.0 VDC (+7 to +15 VDC with wide voltage option)
Pin 2	Rx
Pin 3	Tx
Pin 4	Ground

## 2.2.2 TTL Communications

A standard DB-9F or 4 pin power connector/communication (female) connector is needed for TTL communications. Power may also be supplied via this connector if desired. A PC is not to be used when TTL communication is established via a direct com port. One modification is required for TTL. The RS232 selection jumpers must be removed and placed on the TTL selection jumpers.

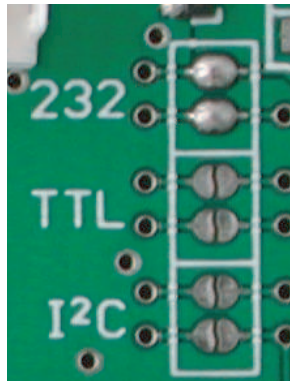


Figure 12: TTL Select

Note that this device uses signal levels from 0V to + 5V on the TTL setting.

Table 6: RS-232 Pinout

Pin Number	Direction	Description	LCD	Host
2	Data from LCD	Data out (LCD)	Tx	Rx
3	Data to LCD	Data in (LCD)	Rx	Tx
5	-	Ground	gnd	gnd

For TTL communication via the 4 pin connector, please see the figure and table below:

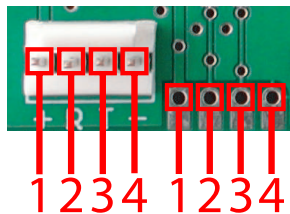


Figure 13: Power Connector

Table 7: Connector Pinout

Pin 1	+5.0 VDC (+7 to +15 VDC with wide voltage option)
Pin 2	Rx (receive data)
Pin 3	Tx (transmit data)
Pin 4	Ground

### 2.2.3 I<sup>2</sup>C Communications

The display I<sup>2</sup>C communications runs at 100 Kbps and supports up to 127 units on a single communications line. The I<sup>2</sup>C data line operates on 5 volt CMOS levels. One modification is required for I2C. The RS232 selection jumpers must be removed and placed on the I2C selection jumpers. Please see figure below for appropriate modification:

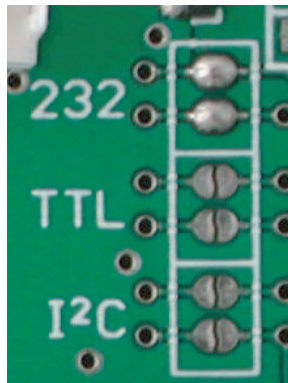


Figure 14: I2C Select

For I<sup>2</sup>C communication, a connection must be made via the 4 pin connector as per the figure and table below:

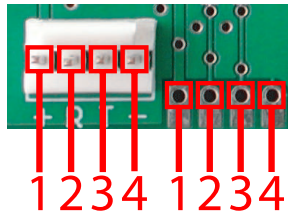


Figure 15: Power Connector



Table 8: Connector Pinout

Pin 1	+5.0 VDC (+7 to +15 VDC with wide voltage option)
Pin 2	I <sup>2</sup> C SDL (clock)
Pin 3	I <sup>2</sup> C SDA(data)
Pin 4	Ground

## 2.2.4 ACK

The idea of ACK is to indicate when the data has been received correctly. ACK does not indicate data incorrectly received. ACK simply fails to indicate when data is correctly received. Clearly, this is of limited usefulness and even less so with Matrix Orbital modules. Matrix orbital modules are not capable of failing to acknowledge an incorrectly received byte in response to that bytes transition. They are only capable of failing to acknowledge the bytes following the byte, which was not received. To fully understand the reasons for this one needs to understand something about how a Matrix Orbital module processes data. Basically the reason why a Matrix Orbital module might fail to receive a byte correctly is that it was unable to process the byte previous before the failed byte was transmitted. Because the module cannot possibly know that it would be unable to store the byte before the next byte was received it cannot know to not ACK. The reason for this situation in deference to situations one might be familiar with (i.e., memory chips, etc) is that the Matrix Orbital module employs a micro-processor to perform these data storage functions. A memory chip takes care of these things entirely within hardware subsystems that operate at the same speed as the transmission themselves.

The display uses a standard Phillips 7bit address as defined by Phillips. However, Matrix Orbital specifies I<sup>2</sup>C address in 8bits. The 8th bit, least significant bit (LSB or Low Order Bit) of the 8bit address is read / write bit. If we take a standard Phillips 7bit address of 45hex this would be in binary 1000101. This is 7bits. Matrix Orbital would describe the Phillips I<sup>2</sup>C address of 45hex as 8Ahex (shifted 1 bit to the left). The read address would be 8Bhex.

For more information on Phillips I<sup>2</sup>C please visit;

<http://www.ping.be/~ping0751/i2cfaq/i2cindex.htm>

## 2.2.5 Dallas 1-Wire Bridge

The display offers one Dallas 1-wire bridge. All three headers are connected to one communication line. A maximum of 32 1-wire devices can be connected to the display at a time.

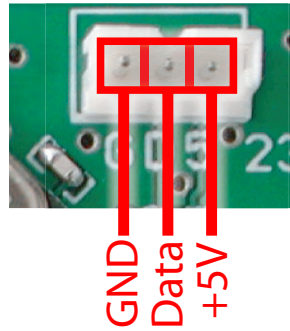


Figure 16: 1-Wire Pinout

## 2.3 General Purpose Outputs

The display has six general purpose outputs. These are provided to control relays or other electronic devices. This allows external devices to be turned on or off using the PC or controller and software commands. Each output is wired as shown in the Figure below. The + terminal is connected directly to the module positive supply, the - terminal is connected through a 240 ohm current limiting resistor and the electronic switch to ground.

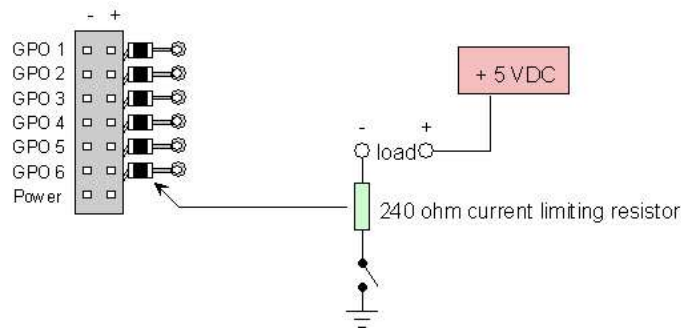


Figure 17: General Purpose Outputs

Maximum allowable current is 20 mA, which is enforced by the current limiting resistor. If the device being switched has a resistance of 240 ohms or more the corresponding resistor may be shorted. Solder a small jumper wire (wirewrap wire is good) from the feedthrough hole to the corresponding negative pin for the GPO in question.

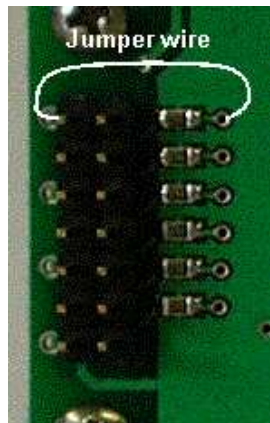


Figure 18: Bypassing 240 ohm Resistor

---

**NOTE** The GPOs do not have any over current or over / under voltage protection so care must be taken when using them. For instance if the external device is a relay it must be fully clamped (using a diode and capacitor) to absorb any generated back electro-motive force (EMF). Suggestions for clamping a relay:

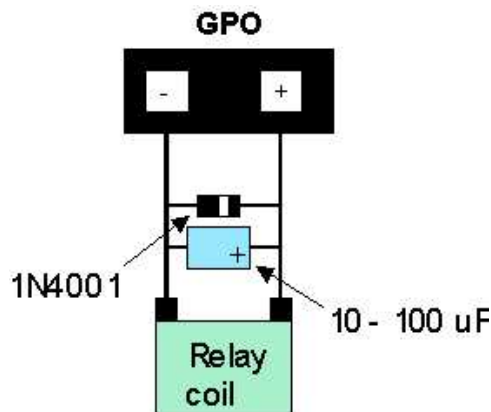


Figure 19: Clamping a Relay

---

### 3 Displaying Text

This chapter describes the various text display commands in detail. **Before issuing commands to the VK202-25 please read sections 8.2 and 8.3 ('Issuing Commands' and 'On Numbers').**

### 3.1 General

Text is displayed on the VK202-25 using the built in 5x7 dot matrix font. In addition, there are up to 8 user defined characters.

### 3.2 The Built In Character Font

The display includes a built in 5x7 dot matrix font with the full range of ASCII characters plus a variety of extended characters, as shown in the Figure below.

D7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
D6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
D5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D3 D2 D1 D0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 0 0 1	1		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 0 1 0	2		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 0 1 1	3		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 1 0 0	4		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 1 0 1	5		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 1 1 0	6		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
0 1 1 1	7		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 0 0 0	8		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 0 0 1	9		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 0 1 0	A		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 0 1 1	B		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 1 0 0	C		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 1 0 1	D		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 1 1 0	E		!	"	\$	%	&	'	(	)	*	+	,	-	.	/
1 1 1 1	F		!	"	\$	%	&	'	(	)	*	+	,	-	.	/

Figure 20: Character Set

In addition to the built in characters, users may define up to 8 special characters which, once defined, occupy positions 0x00 to 0x07 in the above chart. The display does not have provision to download other

fonts.

### **3.3 Writing Text to the Display**

When the display receives a character, it displays that character at the position currently defined. The next character sent to the module then advances to the following position on the display. Characters are drawn using the built in font, and only characters defined in the font are actually displayed. Characters that are not defined by the built in font print as a space (i.e., the cursor is advanced for the next character). The position where text is to be inserted is a character location stored in the display's volatile memory and maintained internally by the display's firmware. This position is manipulated by the commands shown in the following section.

### **3.4 Text Commands**

In this section commands are identified by their names and decimal values. Hex and ASCII equivalents are given in the summary. Some commands, marked with an "R", may be "Remembered" to set new defaults that will be in effect each time the unit is powered on. When "remember" is set/turned on, these commands are automatically remembered to take effect each time the unit is powered on, therefore there is no need to send the save function.

#### **3.4.1 Auto line wrap on (254 67) (R)**

Enables automatic line wrapping. Note that this is not 'word wrapping' and wraps may occur in the middle of a word.

#### **3.4.2 Auto line wrap off (254 68) (R)**

Disables automatic line wrapping. Characters beyond the end of a line will be lost.

#### **3.4.3 Auto scroll on (254 81) (R)**

When auto scrolling is on it causes the display to shift the entire display's contents up to make room for a new line of text when the text reaches the scroll position (the bottom right character position).

#### **3.4.4 Auto scroll off (254 82) (R)**

When auto scrolling is disabled, text will wrap to the top left corner of the display area. Existing text in the display area is not erased before new text is placed. A series of 'spaces' followed by a "Cursor home" command may be used to erase the top line of text.

### **3.4.5 Set cursor position (254 71 [column] [row])**

This command sets the cursor position (text insertion point) to the [column] and [row] specified. Columns have values from 1 to 20 (0x01 to 0x14) and rows have values of 1 and 2 (0x01 and 0x02).

### **3.4.6 Send cursor home (254 72)**

This command moves the cursor position (text insertion point) to the top left of the display area.

### **3.4.7 Turn on underline cursor (254 74) (R)**

Turns on the underline cursor. The cursor shows the current text insertion point. Both underline and blinking cursors may be turned on or off independently. The cursor is off by default.

### **3.4.8 Turn off underline cursor (254 75) (R)**

Turns off the underline cursor. Does not affect the blinking block cursor.

### **3.4.9 Turn on block (blinking) cursor (254 83) (R)**

Turns on the blinking block cursor. The cursor shows the current text insertion point. Both blinking and underline cursors may be turned on or off independently. The cursor is off by default.

### **3.4.10 Turn off block (blinking) cursor (254 84) (R)**

Turns off the blinking block cursor. Does not affect the underline cursor.

### **3.4.11 Cursor left (254 76)**

Moves the cursor one position to the left but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.

---

**NOTE** A 'destructive backspace', which erases the character to the left of the original position, may be done by issuing the following sequence: cursor left, space, cursor left.

---

### **3.4.12 Cursor right (254 77)**

Moves the cursor one position to the right but does not erase any character that may be in that position.

---

**NOTE** This command moves the text insertion point even if the cursor is turned off.

---

## 4 Keypad Interface

This chapter describes the keypad interface and associated commands in detail.

### 4.1 General

The display keypad interface processes the keypad row / column matrix into a serial (RS-232 or I<sup>2</sup>C) data byte stream. Aside from this processing, the keypad has no effect on the display. To send keystrokes to the display they must be routed through the controller.

### 4.2 Connections

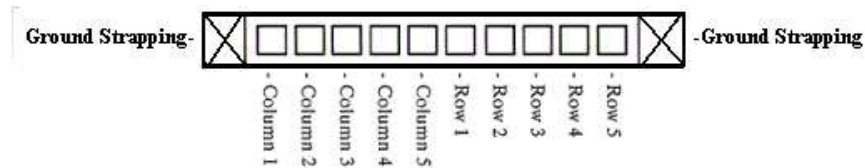


Figure 21: Keypad Connector

The connector is not 'keyed' so the keypad will probably plug in either of two ways. The display will not be damaged by reversing the connector, but the keypad will generate a different ASCII character mapping for each position. If the connector has fewer than 10 pins it should be centered on the display connector.

The diagram shows the logical layout (row 1, column 1 in upper left). The connector for the keypad is a 10 pin 0.1" spacing male header. Pins 1 through 5 are columns and pins 6 through 10 are rows. The keypad is scanned whenever a key is pressed; there is no continuous key scan. This means that key presses are dealt with immediately without any appreciable latency. This also prevents electrical noise which is often caused by continuous key scans.

---

**NOTE** The keypads may be laid out in a different pattern. If this is the case, the user will need to interpret the key codes differently. Also included are two extra pins on each end of the connector to be used for ground strapping. This can be used in conjunction with your keypad if a ground strap connection is required or if a common ground connection is needed.

---

Table 9: Keypad Layout

	Columns					
	1	2	3	4	5	
	1	A	B	C	D	E
Rows	2	F	G	H	I	J
	3	K	L	M	N	O
	4	P	Q	R	S	T
	5	U	V	W	X	Y

---

**NOTE** The keypad connector must be wired with columns on one side and rows on the other side of the center of the connector. In situations where the keypad isn't wired this way an adapter will need to be made, or the user should rewire the connector to meet this requirement.

---

### 4.3 I<sup>2</sup>C Interface

The keypad can be read by I<sup>2</sup>C master read. This means that a read of the module will always return the first unread key press. A read is initiated by writing to the module with its base address plus 1, then clocking the module's return byte after the module releases the SDA line. Much more detail on this basic I<sup>2</sup>C function can be found in the I<sup>2</sup>C specification by Phillips.

A good reference is also available at:

<http://www.ping.be/~ping0751/i2cfaq/i2cindex.htm>

The module contains a ten key press buffer so that it can be polled for key presses at an infrequent rate (every 500 to 1000 ms is typical). All returned key presses indicate the presence or absence of additional logged key presses by the most significant bit (MSB - bit 7). If the user has pressed two keys since the last poll of the keypad interface, the first read will return the key code with bit 7 set and the second read will return the key code with bit 7 clear. The application must take into account this bit to keep up with user key presses. If there are no keypresses detected, the module will return zero (0x00).

### 4.4 RS-232 Interface

By default on any press of a key, the module will immediately send out the key code at the selected baud rate. This behavior can be modified using commands found in the next section.

### 4.5 Keypad Commands

Some commands, marked with an "R", may be "Remembered" to set new defaults that will be in effect each time the unit is powered on. When "remeber" is set / turned on, these commands are automatically remembered to take effect each time the unit is powered on, therefore there is no need to send the save



function.

### 4.5.1 Auto repeat mode on (254 126 [mode]) (R)

[mode] = 0x00 gives Resend Key Code mode

[mode] = 0x01 gives Key Down / Key Up Code mode

1. **Resend Key Code:** This mode is similar to the action of a keyboard on a PC. In this mode, when a key is held down, the key code is transmitted immediately followed by a 1/2 second delay. After this delay, key codes will be sent via the RS-232 interface at a rate of about 5 codes per second. This mode has no effect if polling or if using the I<sup>2</sup>C interface.
2. **Key Down / Key Up Codes:** This mode may be used when the typematic parameters of the “Resend key code” mode are unacceptable or if the unit is being operated in polled mode. The host system detects the press of a key and simulates an auto repeat inside the host system until the key release is detected. In this mode, when a key is held down, the key code is transmitted immediately and no other codes will be sent until the key is released. On the release of the key, the key release code transmitted will be a value equal to the key down code plus 20 hex. For example, the key code associated with key 'P' (0x50) is pressed, the release code is 'p' (0x70). In RS-232 polled mode or via the I<sup>2</sup>C interface, the “Key down / key up codes” are used. However, the user should be careful of timing details. If the poll rate is slower than the simulated auto-repeat it is possible that polling for a key up code will be delayed long enough for an unwanted key repeat to be generated.

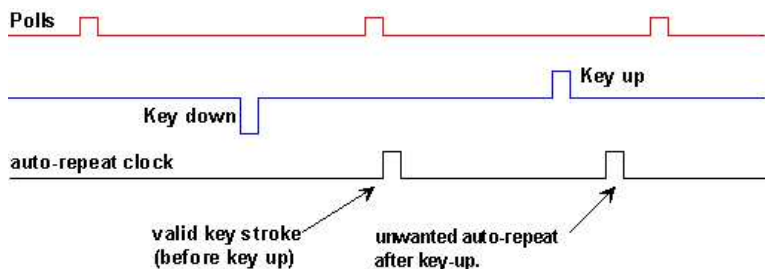


Figure 22: Poll Timing

### 4.5.2 Auto repeat mode off (254 96) (R)

This command turns off auto repeat mode.

### 4.5.3 Auto transmit keypresses on (254 65) (R)

In this mode, all keypresses are sent immediately to the host system without the use of the poll keypad command. This is the default mode on power up.

#### **4.5.4 Auto transmit keypresses off (254 79) (R)**

In this mode, up to 10 keypresses are buffered until the unit is polled by the host system via the poll keypad command. Issuing this command places the unit in polled mode.

#### **4.5.5 Clear key buffer (254 69)**

This command clears any unread keypresses. In a menuing application, if the user presses a key which changes the menu context, any following key presses may be inaccurate and can be cleared out of the buffer between menu changes to prevent jumping around the menu tree. It may also be used to, in effect, reset the keypad in case the host application resets for whatever reason.

#### **4.5.6 Poll keypad (254 38)**

This command returns any unbuffered keypresses via the RS-232 interface. The host system must be set up to receive the key codes. When the display receives this command it will immediately return any unbuffered keypresses which may have not been read already. If there is more than one keypress buffered, then the high order bit (MSB) of this returned keycode will be set (1). If this is the only buffered keypress, then the MSB will be reset (0). If there are no buffered keypresses, then the returned code will be 0x00. Please note to make use of this command the "Auto transmit keypress" mode should be off.

#### **4.5.7 Set debounce time (254 85 [time]) (R)**

[time] is in increments of 6.554 milliseconds.

This command sets the time between key press and key read. All key types with the exception of latched piezo switches will 'bounce' for a varying time, depending on their physical characteristics. The default debounce time for the module is about 52 ms, which is adequate for most membrane keypads.

## **5 Bar Graphs and Special Characters**

The display includes the ability to draw bar graphs (either horizontal or vertical) and allows users to define up to eight special characters.

Eight characters (ASCII values 0x00 to 0x07) are set aside for use with bar graphs, user defined characters, and big numbers. Since the same 8 characters are used for each function, the functions may not be used simultaneously. The bottom row of pixels is normally reserved for the underline cursor. The characters may be defined or redefined at any time by issuing the commands shown in this section. Once defined, they may be used either by means of the bar graph commands, or by simply issuing one of the ASCII values 0x00 to 0x07 (which is not prefixed by the command byte, 254).

## 5.1 Command List

### 5.1.1 Initialize wide vertical bar graph (254 118)

This command defines the 8 special / user characters to be blocks suitable for use in drawing wide (5 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are redefined by another command.

### 5.1.2 Initialize narrow vertical bar graph (254 115)

This command defines the 8 special / user characters to be blocks suitable for use in drawing narrow (2 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are redefined by another command.

### 5.1.3 Draw vertical bar graph (254 61 [column] [height])

Draws a vertical bar graph in [column] having a height of [height] pixels. The height may range from 0 to 20 (0x00 to 0x14) pixels. The necessary characters must first be initialized by either of the commands shown in section 5.1.1 or 5.1.2, which will determine the width of the graph drawn. Graph may be erased by drawing a bar graph of height = 0 in the same column.

### 5.1.4 Initialize horizontal bar graph (254 104)

This command defines the 8 special / user characters to be blocks suitable for use in drawing horizontal bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of horizontal bar graphs may be drawn unless the characters are redefined by another command.

### 5.1.5 Draw horizontal bar graph (254 124 [column] [row] [dir] [length])

Draws a horizontal bar graph in [row] starting at [column] with a length of [length] pixels. [row] may have a value of 0x01 or 0x02, column may range from 0x01 to 0x14 and length may be from 0x00 to 0x64 (0 to 100) if the graph can extend the full width of the screen. Each column is 5 pixels wide (spaces between the columns don't count).

[dir] specifies the direction: 0x00 goes from left to right, 0x01 goes from right to left.

### 5.1.6 Define custom character (254 78 [c] [8 bytes])

The display allows up to 8 user defined (custom) characters. These characters occupy the first 8 (0x00 to 0x07) places in the character set.

Custom characters occupy a 5x7 pixel matrix. Built in characters are 5x7: the bottom row of pixels is normally reserved for the underline cursor. The underline cursor should be turned off if the bottom row of pixels forms part of a custom character.

The characters are defined by issuing the command: 254 78 [c], followed by 8 bytes to define the character. [c] is the character number (0x00 to 0x07). The 8 bytes are mapped as shown below;

Table 10: 8 Byte Map

MSB LSB								
*	*	*	1	2	3	4	5	Data Byte 1
*	*	*	6	7	8	9	10	Data Byte 2
*	*	*	11	12	13	14	15	Data Byte 3
*	*	*	16	17	18	19	20	Data Byte 4
*	*	*	21	22	23	24	25	Data Byte 5
*	*	*	26	27	28	29	30	Data Byte 6
*	*	*	31	32	33	34	35	Data Byte 7

A "1" bit indicates an on (black) pixel, a "0" bit indicates an off (clear) pixel.

Once defined, a character is displayed simply by issuing a value (0x00 to 0x07) corresponding to the character number. The character will be laid out as follows;

Table 11: Character Values

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
<b>Cursor Line</b>				

---

**NOTE** Custom characters will be erased if any of the "Initialize bar graph" commands are issued.

---

### 5.1.7 Initialize Medium Digits (254 109)

This command defines the 8 special characters to be blocks suitable for use in drawing medium digits. Any previous definitions will be lost.

Once this command has been issued, any number of medium characters may be placed until the characters are redefined by another command.

### 5.1.8 Draw Medium Digits (254 111 [row][column][digit])

Draws a medium digit in [row] and [column] using the specified [digit]. Medium digits occupy two rows. When the [row] is specified to be "1", the medium digit will be displayed correctly. When the [row] is specified to be "2", only the top part of the medium digit will be displayed. When [row] is specified to be "0", only the bottom part of the medium digit will be displayed.

## 6 Miscellaneous Commands

The commands listed in this chapter don't readily fit in any of the other categories, or are used in more than one category.

### 6.1 Command List

Some commands, marked with an "R", may be "Remembered" to set new defaults that will be in effect each time the unit is powered on. When "remember" is set / turned on, these commands are automatically remembered to take effect each time the unit is powered on, therefore there is no need to send the save function.

#### 6.1.1 Remember (254 147 [0|1])

This command allows a number of settings, such as cursor state, backlight, etc., to automatically be stored to non-volatile memory so they become new defaults. The command should only be used when required for two reasons;

1. Writing to non-volatile memory ( using the "save" function) is time consuming and slows down the operation of the display.
2. Non-volatile memory has a 'write limit' and may only be changed approximately 100,000 times.

#### 6.1.2 Clear display (254 88)

This command clears the display and resets the text insertion point to the top left of the screen.

### 6.1.3 Set brightness (254 89 [brightness]) (R)

This command sets the display's brightness to [brightness], where [brightness] is a value between 0x00 and 0x03 (between 0 and 3) according to the table below.

Table 12: Brightness Settings

Hex Values	Brightness
0x00	25%
0x01	50%
0x02	75%
0x03	100%

Lighting conditions will affect the actual value used for optimal viewing. Brightness is set to 100% by default. Brightness cannot be 'remembered' and must be set each time a change is required.

### 6.1.4 Set brightness and Save (254 145 [brightness]) (R)

This command sets the display's brightness to [brightness] and saves it on subsequent power ups, where [brightness] is a value between 0x00 and 0x03 (between 0 and 3) according to the table, "Brightness Settings" as seen above.

### 6.1.5 Display on (254 66 [minutes]) (R)

This command turns on the display for a time of [minutes] minutes. The maximum value of [minutes] is 100. If [minutes] is zero (0), the display will remain on indefinitely.

### 6.1.6 Display off (254 70) (R)

This command turns the display off.

### 6.1.7 Load startup screen (254 64 [40 characters])

This command sets and memorizes the startup screen that will appear each time the display is turned on. By default the screen shows:

Table 13: Default Screen

Matrix Orbital
VK202-25

The 40 characters define the two 20 character rows of the screen.

Table 14: Default Screen

Character 1	Character 20
Character 21	Character 40

If sending more than 10 characters to be stored, add in a ~10ms per character delay. Predefined custom characters can be used in the “Start up Screen” as well, by using 0x00 through 0x07 characters.

### 6.1.8 General purpose output off (254 86 [gpo #])

This command turns OFF any of the general purpose outputs. [gpo #] is 1 to 6. Note that OFF means that the output is high.

### 6.1.9 General purpose output on (254 87 [gpo #])

This command turns ON any of the general purpose outputs. [gpo #] is 1 to 6. ON means that the output is pulled low (ground via 240 ohms).

### 6.1.10 Remember GPO State (254 195 [gpo#][gpo value])

This command will set the startup state of all the GPO’s. When the device is powered up the next time, the GPO’s will be set to the values from this command. [gpo#] is referenced to the GPO in which you select to remember. [gpo value] is referenced to the state of the gpo that is to be remembered. “1” being “on” and “0” being “off”.

### 6.1.11 Set I<sup>2</sup>C address (254 51 [address]) (R)

This command sets the I<sup>2</sup>C write address of the module. This value must be an even number and the read address is one higher. For example, if the I<sup>2</sup>C write address is set to 0x50, then the read address is 0x51. The

change in address is immediate. This address is 0x50 by default, and is reset temporarily back to that value when the 'manual over-ride' jumper is used on power up.

### 6.1.12 Read module type (254 55)

This command will return a 1 byte hex value, over the RS-232 interface only, the model type value of the module. Values for various modules at the time of this publication are as follows:

Table 15: Module Values

LCD0821 - 0x01	LCD2021 - 0x03	LCD2041 - 0x05
LCD4021 - 0x06	LCD4041 - 0x07	LK202-25 - 0x08
LK204-25 - 0x09	LK404-55 - 0x0A	VFD2021 - 0x0B
VFD2041 - 0x0C	LK162-12 - 0x34	<b>VK202-25 - 0x0E</b>
VK204-25 - 0x0F	GLC12232 - 0x10	GLC24064 - 0x13
GLK24064-25 - 0x15	GLK12232-25 - 0x22	LK402-12 - 0x33

### 6.1.13 Set RS-232 port speed (254 57 [speed]) (R)

This command sets the display's RS-232 port to the specified [speed]. The change takes place immediately. [speed] is a single byte specifying the desired port speed. Valid speeds are shown in the table below. The display can be manually reset to 19,200 baud in the event of an error during transmission (including transmitting a value not listed below) by setting the 'manual over-ride' jumper on the display controller board during power up. This command is ignored until this jumper is removed again.

Table 16: Speed Settings

Speed Value	Speed
FF Hex	1200 baud
81 Hex	2400 baud
20 Hex	9600 baud
0F Hex	19200 baud

### 6.1.14 Set Serial Number (254 52 [byte1] [byte2])

Modules may be delivered with the serial number blank. In this case the user may set the desired 2 byte serial number using this one time only command.

Upon the execution of this command, the module will echo these two bytes back over the RS-232 interface. The serial number may be set only once. Any future attempt to execute this command will result in no change and the module will return to the originally set serial number.



### 6.1.15 Read Serial Number (254 53)

This command will return a 2 byte hex value, over the RS-232 interface only, the serial number of the module as it was previously stored.

### 6.1.16 Read Version Number (254 54)

This command will return the firmware version number of the display as a 1 byte hex value over RS-232 only.

## 6.2 Flow Control

The display has built in flow control which may be useful when long strings of text are downloaded to the display. Flow control is enabled or disabled by two commands. If flow control is enabled, the display will return an "almost full" message (0xFE) to the controller when its internal buffer fills to a defined level, and an "almost empty" message (0xFF) when the buffer contents drop to a defined level.

### 6.2.1 Enter Flow Control Mode (254 58 [full][empty])

---

**NOTE** Flow control applies only to the RS-232 interface. It is not available for I<sup>2</sup>C.

---

This command enables flow control. When the buffer fills so that only [full] bytes are available the display will return an "almost full" message (0xFE) to the controller. When the buffer empties so that only [empty] bytes remain the display will return an "almost empty" message (0xFF) to the controller.

The display will return the "almost full" message for every byte sent to the display until the used buffer space once more drops below the [full] level.

Whether the user is in 'flow control mode' or not, the module will ignore display or command bytes which would overrun the buffer. While in 'flow control mode' the unit will return 0xFE when buffer is almost full even though it may have already thrown rejected data away. The buffer size for the display is 80 bytes.

When using this command in an application, selection of the value for the buffer almost full should be considered very carefully. This is a critical aspect of using this command to it's full potential. When using a host system or PC which contains a FIFO, the user should set the value of [full] equal to or greater than the size of the FIFO. The reason for this is that the FIFO may be full when the host system receives 0xFE. In the case of 16550 UART the size at its maximum is 16, therefore the value [full] should be set to 16 or greater.

### 6.2.2 Exit Flow Control Mode (254 59)

This command turns off flow control. Bytes may overflow the buffer without warning.

## 7 1-Wire Commands

The 1-wire bus is capable of communicating with many devices over a single wire plus a ground reference. This chapter deals with the capabilities of the display and a brief introduction to the 1-Wire standard. For more detail consult [www.maxim-ic.com](http://www.maxim-ic.com)

### 7.1 Device Identification

Each 1-wire device contains a unique 64-bit address in which to identify them with. The address is guaranteed to be unique from any other device, allowing a virtually unlimited number of devices on to be attached to the bus. The address itself contains a family code and a cyclic redundancy check (CRC). The family code is unique to a particular device model. For example, the family code for the DS18S20 temperature probes is 10H. The CRC byte is included as a verification that the correct address was transmitted or received.

### 7.2 Protocol

The transaction sequence for accessing a 1-wire device is as follows:

1. Reset and detect presence.
2. ROM command, followed by any required data exchange.
3. Device specific function command, followed by any required data exchange.

Before communication can begin, the bus must be reset to force all devices to begin listening. After the reset, all the devices will transmit a presence pulse which indicates that there is at least one device on the bus. Once the presence of at least one device has been confirmed, the master must select which group of devices will be involved in the rest of the transaction. A device will use a ROM command to determine if the following transaction is intended for it. If not, the device will ignore all communication on the bus until the next bus reset. Otherwise, the device will read and process the rest of the transaction. The final part of the transaction is the device specific function command. To determine what functions the device will respond to, consult the devices' data sheet.

It is very important to follow this sequence for every transaction. If any of these steps are omitted or performed in the wrong order, the devices will not respond.

None of the device addresses are known ahead of time, and as such, each of their addresses must be searched out and determined. The 1-wire bus provides a means of searching the bus for devices and determining their address.

### 7.3 ROM Commands

The ROM commands allow a device to be singled out for communication or all devices to be included. This manual only presents the three most used ROM commands. For a more detailed listing and description of all the ROM commands, consult the data sheet for the 1-wire device being used.

- **Match ROM [55h]:** To single out a device, the Match ROM command is used. After this command has been issued the 64-bit target address is transmitted in LSB to MSB order. Any device with an address that doesn't match will ignore all further communication until the next bus reset.
- **Skip ROM [CCh]:** After this command, all devices will continue to listen and process the transaction. This is equivalent to broadcasting to all devices. This command is useful when probes need to be informed to get their measurement ready. With this command all the probes can be instructed simultaneously.
- **Read ROM [33h]:** All devices will begin transmitting their address after this command. This command will only succeed when there is one device on the bus. When multiple devices are present, all devices will begin transmitting their addresses, overlapping each other. This command can be used to determine if there is more than one device on the bus. After the address has been read back, if the CRC is valid, there is only one device on the bus. Otherwise, there are multiple devices on the bus and each address must be searched out.

## 7.4 Display 1-Wire Connections

The transaction command allows data to be put onto the bus and read off the bus for transactions. And the search command identifies all the devices on the bus for further communication.

### 7.4.1 Transaction command (254 C8 1 [flags] [Send Bits] [Recieve bits] [Send data])

The transaction command will perform a single transaction on the 1-wire bus in this order;

1. Bus Reset.
2. Transmit data onto the bus.
3. Receive data from the bus.

Table 17: 1-Wire Transaction

Offset (Bytes)	Length (Bytes)	Name	Description
0	1	Flags	The flags byte controls the optional components of the transaction.
1	1	Send Bits	The number of bits that will be transmitted onto the bus. The actual bits to be transmitted are held in the Send Data section.
2	1	Recieve Bits	The number of bits to read off the bus after the data to be put on the bus has been sent.
3	Variable	Send Data	The data to be transmitted onto the bus. The data is transmitted MSB to LSB in the order that they are received.

Table 18: 1-Wire Flags

Bit	Description
7	
6	Unused
5	(0 for future compatibility)
4	
3	Add a CRC8 to the end of the transmitted data
2	(0 for future compatibility)
1	Assume last received byte is a CRC8 and validate it
0	Reset bus before transaction

The received data is sent back using the “Display return protocol”.

Table 19: 1-Wire Error Codes

Code	Description
0x00	Success
0x01	Unknown 1-Wire Command
0x02	No devices on the bus
0x03	Fatal search error

## 7.4.2 Search command (254 200 2)

This is used to find the addresses of all 1-Wire devices on the bus. After this command the display will return one or more “Display return protocol” packets containing either an error code or addresses of 1-wire packets.

Table 20: Search Return Packet

Offset (Bytes)	Offset (Bytes)	Description
0	2	0x232A - Preamble
2	1	0x8A - Packet is 10 bytes long, another address will follow 0x0A - Packet is 10 bytes long, this is the last address
3	1	0x31 - 1-Wire Packet Type
4	1	Error Code (0x00 for success)
5	8	1-Wire Address
13	1	CRC8 - 0x00 means the last address was valid

# 8 Appendix: Command Summary

## 8.1 General

The operation of the display is controlled by a simple and consistent command set. Commands control;

- Text display
- Graphics display
- Keypad interface
- Miscellaneous operating
- Parameters

This chapter includes summary tables of all commands.

## 8.2 Issuing Commands

Commands are issued to the display by the controller. In a test setup, commands can be issued to the display by means of a BASIC program, using the chr\$( ) function. In the tables below, we've shown commands in hex, ASCII and decimal form. All commands begin with the prefix character 0xFE (254 decimal). These commands are issued on the serial communications link (I<sup>2</sup>C or RS-232) at the currently defined baud rate.

For example (using BASIC in a test setup), the user could issue the command to clear the screen on the display by including the line;

```
PRINT#1, chr$(254);chr$(88)
```

In the BASIC program.

Or, with C the user could (using Zcomm serial library)

```
ZComm1->WriteCommByte(0xfe);  
ZComm1->WriteCommByte('X');
```

## 8.3 On Numbers

Like all computerized devices, the display operates with commands and values in the form of binary numbers. These binary numbers are arranged in 8 digit (i.e., 8 bit) groups called bytes. The decimal value of a byte may have any value from 0 to 255.

Bytes are usually specified in either decimal or hexadecimal (base 16) form for convenience, since binary numbers are confusing to deal with directly. Hexadecimal (hex) numbers are particularly convenient because exactly two hexadecimal digits make up one byte, each hex digit representing 4 binary digits (4 bits) as shown here;

Table 21: Hex Value Table

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Based on the table, the byte 01001011 can be represented in hex as 4B, which is usually written as any of 4Bh, 4BH, 4B hex or 0x4B.

The numbers can also be expressed in decimal form if preferred.

### 8.3.1 ASCII Characters

Since computers deal internally with numbers only, but externally with both letters and numbers, several schemes were developed to 'map' written characters to numeric values. One such scheme has become universal, the American Standard Code for Information Interchange, or ASCII. ASCII tables are readily available from a number of sources. A few examples will do here:

Table 22: Example of an ASCII Table

The letter	A	has a value of	65 Decimal or	41 Hex
The letter	A	has a value of	97 Decimal or	61 Hex
The number	0	has a value of	48 Decimal or	30 Hex
The number	9	has a value of	57 Decimal or	39 Hex

This gives rise to the possibility of confusion when parameters are being set on the display. For example, the GPO ON and OFF commands use a number to indicate which GPO is being controlled. We're told that acceptable values are 1 to 6. **All such parameters must use numeric values (i.e., the actual byte values).** If we send the ASCII number 0 by mistake it will actually give the value 48 decimal (30 hex) to the parameter, which is wrong.

In the tables given in the following sections ASCII characters are shown as 'A', with single quotes.

## 8.4 Text Commands

Syntax in the tables below are given in hex, decimal, and decimal with ASCII.

Table 23: Text Commands

Command	Syntax	Default	Notes
Auto line wrap on	FE 43 254 67 254 'C'	On R	Enables line wrapping (not word wrap).
Auto line wrap off	FE 44 254 68 254 'D'	On R	Disables line wrapping.
Auto scroll on	FE 51 254 81 254 'Q'	Off R	Enables scroll at bottom of screen. Text will push display up one line to make room for new line.
Auto scroll off	FE 52 254 82 254 'R'	Off R	Disables auto scroll. Text will wrap to top left and overwrite existing text

<b>Command</b>	<b>Syntax</b>	<b>Default</b>	<b>Notes</b>
Set cursor position	FE 47 [col] [row] 254 71 [col] [row] 254 'G' [col] [row]	N/A	Moves cursor to the specified column and row. The cursor marks the text insertion point in this and all commands.
Send cursor home	FE 48 254 72 254 'H'	N/A	This command moves the cursor to the top left of the display area.
Underline cursor on	FE 4A 254 74 254 'J'	Off R	Turns on the underline cursor.
Underline cursor off	FE 4B 254 75 254 'K'	Off R	Turns off the underline cursor.
Block cursor on	FE 53 254 83 254 'S'	Off R	Turns on the blinking block cursor.
Block cursor off	FE 54 254 84 254 'T'	Off R	Turns off the blinking block cursor.
Cursor left	FE 4C 254 76 254 'L'	N/A	Moves the cursor one position to the left. If the cursor is already at the beginning of a line it will move to the end of the other line.
Cursor right	FE 4D 254 77 254 'M'	N/A	Moves the cursor one position to the right. If the cursor is already at the end of a line it will move to the beginning of the other line.



## 8.5 Keypad Interface Commands

Table 25: Keypad Interface Commands

Command	Syntax	Default	Notes
Auto repeat mode on	FE 7E [0x00 0x01] 254 126 [011] 254 '~' [011]	Off R	Applies to keypad only. 0x00 = 200 ms typematic, 0x01 = key down/key up codes sent.
Auto repeat mode off	FE 60 254 96 254 '^^'	Off R	Applies to keypad only.
Auto transmit key presses on	FE 41 254 65 254 'A'	On R	Sets auto transmit mode for keypad. Key presses are transmitted to host without polling.
Auto transmit key presses off	FE 4F 254 79 254 'O'	Off R	Up to 10 key presses buffered until polled.
Clear key buffer	FE 45 254 69 254 'E'	N/A	Clear unread key presses.
Poll keypad	FE 26 254 38 254 '&'	N/A	Returns buffered key presses to application. Returns 0x00 if no key presses. High order bit set unless this is the last / only key press.
Set debounce time	FE 55 [time] 254 85 [time] 254 'U' [time]	52ms R	Resolution: 1 = 0.6554 ms [time] is a numeric multiplier.

## 8.6 Bar Graphs and Special Characters

The commands in this section are used to define and display bar graphs and special characters.

Table 27: Bar Graphs and Special Characters

Command	Syntax	Notes
Initialize thick vertical bar graph	FE 76 254 118 254 'v'	Initializes the user character set to make wide vertical bar graphs.
Initialize thin vertical bar graph	FE 73 254 115 254 's'	Initializes the user character set to make narrow vertical bar graphs.
Initialize medium digits	FE 6D 254 109 254 'm'	Initializes the character set to make medium digits.
Initialize horizontal bar graph	FE 68 254 104 254 'h'	Initializes the user character set to make horizontal bar graphs.
Define custom character	FE 4E [c][8 bytes] 254 78 [c][8 bytes] 254 'N' [c][8 bytes]	Defines one of 8 custom "user" characters. Character number is [c] between 0x00 and 0x07.
Draw vertical bar graph	FE 3D [col][length] 254 61 [col][length] 254 '= ' [col][length]	Draws a vertical bar graph at column [col] of length [length]. Length is measured in pixels (0x00 to 0x14). User must first use the 'v' or 's' command to initialize characters.
Draw horizontal bar graph	FE 7C [c][r][d][length ] 254 124 [c][r][d][length] 254 'l' [c][r][d][length]	Draws a horizontal bar graph starting at column [c] on row [r] with direction [d] (0 is right, 1 is left) of length [length]. Length is measured in pixels (0x00 to 0x64 if starting in column 1). User must first use the 'h' command to initialize characters.

Command	Syntax	Notes
Draw medium digits	FE 6F [r][c][digit] 254 111 [r][c][digit] 254 'o' [r][c][digit]	Draws a medium digit in [row] and [column] using the specified [digit]. Medium digits occupy two rows. When the [row] is specified to be "1", the medium digit will be displayed correctly. When the [row] is specified to be "2", only the top part of the medium digit will be displayed. When [row] is specified to be "0", only the bottom part of the medium digit will be displayed.

## 8.7 Miscellaneous Commands

Table 29: Miscellaneous Commands

Command	Syntax	Default	Notes
Remember	FE 93 [011] 254 147	Off	Turn the "remember" function on [1] or off [0].
Clear display	FE 58 254 88 254 'X'	N/A	Clears screen of text and graphics, places text cursor at top left.
Set brightness	FE 59 [brightness] 254 89 [brightness] 254 'Y' [brightness]	0x00 R	Sets display brightness. Brightness is a value between 0x00 and 0x03. 0x00=25%, 0x01=50%, 0x02=75%, 0x03=100%.

Command	Syntax	Default	Notes
Remember	FE 93 [01] 254 147	Off	Turn the “remember” function on [1] or off [0].
Set brightness and save	FE 91 [brightness] 245 145 [brightness]	0x03 R	Sets display brightness and saves on subsequent power ups. Brightness is a value between 0x00 and 0x03. 0x00=25%, 0x01=50%, 0x02=75%, 0x03=100%.
Display on	FE 42 [minutes] 254 66 [minutes] 254 'B' [minutes]	On R	Display will stay on for [minutes]. If [minutes] = 0 display will stay on permanently. Maximum value for [minutes] is 100.
Display off	FE 46 254 70 254 'F'	Off R	Turns off display.
Load startup screen	FE 40 [40 char] 254 64 [40 char] 254 '@' [40 char]	Matrix Orbital VK202-25	Loads new startup screen (40 characters). Screen is remembered for subsequent power ups.
General purpose output off	FE 56 [gpo #] 254 86 [gpo #] 254 'V' [gpo #]	Off R	Turns a general purpose output OFF. [gpo #] may be from 1 to 6.
General purpose output on	FE 57 [gpo #] 254 87 [gpo #] 254 'W' [gpo #]	Off R	Turns a general purpose output ON. [gpo #] may be from 1 to 6.
Remember general purpose output	FE C3 [gpo#][gpo value] 254 195 [gpo#][gpo value]	Off	Remembers the state of each GPO after power up.

Command	Syntax	Default	Notes
Remember	FE 93 [01] 254 147	Off	Turn the "remember" function on [1] or off [0].
Set I <sup>2</sup> C address	FE 33 [address] 254 51 [address] 254 '3' [address]	0x50	Value is write address and must be even, read address is 1 higher.
Read module type	FE 37 254 55 254 '7'	see table	Reads the module type.
Set RS-232 port speed	FE 39 [speed] 254 57 [speed] 254 '9' [speed]	19,200 baud	Sets RS-232 speed.
Set Serial Number	FE 34 [byte1][byte2] 254 52 [byte1][byte2] 254 '4' [byte1][byte2]	N/A	This is a one-time-only command which works only on units without factory set serial numbers.
Read Serial Number	FE 35 254 53 254 '5'	N/A	Reads the two byte serial number of the module.
Read Version Number	FE 36 254 54 254 '6'	N/A	Reads the firmware version number of the module.
Enter flow control mode	FE 3A [full] [empty] 254 58 [full] [empty] 254 ':' [full] [empty]	Off R	Sets "full" and "empty" marks for the 80 byte display buffer. When buffer reaches [full] display will return 0xFE to host. When buffer reaches [empty] display will return 0xFF.
Exit flow control mode	FE 3B 254 59 254 ';' ;'	Off R	Turns off flow control.

## 9 Appendix: Specifications and Options

### 9.1 Specifications

Table 31: Environmental Specifications

	Standard Temperature	Extended Temperature
Operating Temperature	-20°C to +70°C	-40°C to +85°C
Storage Temperature	-40°C to +80°C	-50°C to +85°C
Operating Relative Humidity	20 to 80% non condensing	
Vibration (non-operating)	10 to 55 to 10 Hz (Frequency)	
	1.0 mm (Total Amplitudes)	
	30 minutes (Duration)	
	X, Y, Z each direction	
Shock (Non-operation)	593 m/S <sup>2</sup> , 10 mS	

Table 32: Electrical Specifications

Supply Voltage	4.75 - 5.25 Vdc (optional 8-15 VDC)
Supply Current (all dots on)	130 mA typical
Supply Current (power off)	9 mA typical
Supply Current (power up)	Up to 250 mA inrush

Table 33: Optical Characteristics

Number of Characters	40 (20 characters by 2 lines)
Matrix format	5 x 7 with underline
Display Area	70.8 x 11.5 mm XxY
Character Size	2.4 x 4.7 mm (XxY), not including underline
Character Pitch	3.6 mm
Line pitch	6.1 mm
Dot Size	0.4 x 0.5 mm (XxY)
Dot Pitch	0.5 x 0.7 mm (XxY)
Luminance	350cd / m <sup>2</sup> (100fL) min
Color of Illumination	Blueish Green

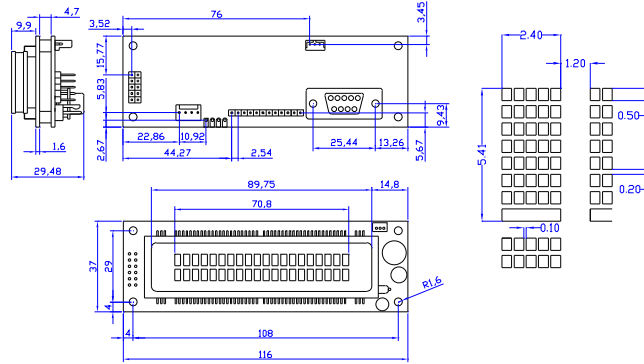


Figure 23: Physical Layout

## 9.2 Options

Table 34: Options Available on VK202-25

Extended Temperature	E
Wide Voltage	V

## 10 Appendix: Glossary

Table 35: Appendix: Glossary

<b>ASCII</b>	American Standard Code for Information Interchange. A 7 bit binary code representing the English alphabet, decimal numbers and common punctuation marks. Also includes control characters such as carriage return or end of text. An 8 bit superset of the standard ASCII codes is often used today to include foreign characters and other symbols. These supersets are often called extended ASCII character sets.
--------------	--

<b>Backlight</b>	A backlit display is illuminated from behind to provide nighttime and improved daytime readability.
<b>Binary Number</b>	The (data and signaling) bit transmission rate of an RS-232 device.
<b>Bit</b>	A number written using binary notation which only uses zeros and ones.
<b>Bitmap</b>	A representation, consisting of rows and columns of dots, of a graphics image in computer memory. The value of each dot (whether it is filled in or not) is stored in one or more bits of data.
<b>Byte</b>	A grouping of eight binary bits
<b>CCFL</b>	Cold Cathode Fluorescent Lamp. A high brightness backlighting source consists of a fluorescent tube powered by a high voltage A.C. source.
<b>Configuration</b>	The way a system is set up, or the assortment of components that make up the system. Configuration can refer to either hardware or software, or the combination of both.
<b>Contrast</b>	The ratio of luminance between the light state of the display to the dark state of the display.
<b>Controller</b>	The micro-controller or PC used to control the Matrix Orbital display unit.
<b>DB-9</b>	The designation of a connector used in the RS-232 interface: 9 pin connector
<b>Firmware</b>	Software (programs or data) that has been written onto read-only memory (ROM). Firmware is a combination of software and hardware. ROMs, PROMs and EPROMs and flash EEPROMs that have data or programs recorded on them are firmware.
<b>Font</b>	A design for a set of characters. A font is the combination of typeface and other qualities, such as size, pitch, and spacing.
<b>Font Metric</b>	A definition of where font is to be placed, such as margins and spacing between characters and lines.
<b>Hexadecimal</b>	Refers to the base-16 number system, which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F. For example, the decimal number 15 is represented as F in the hexadecimal numbering system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for humans to read hexadecimal numbers than binary numbers.



<b>I<sup>2</sup>C</b>	Short for Inter-IC, a type of bus designed by Phillips Semiconductors in the early 1980s, which is used to connect integrated circuits (ICs). I <sup>2</sup> C is a multi-master bus, which means that multiple chips can be connected to the same bus and each one can act as a master by initiating a data transfer.
<b>Interface</b>	A means by which two systems interact.
<b>LCD</b>	Liquid Crystal Display
<b>Module Type Value</b>	This refers to the model number of the module.
<b>Pixel</b>	The smallest individually controllable element of a display.
<b>PLED</b>	Polymer Light Emitting Diode. Polymers are substances formed by a chemical reaction in which two or more molecules combine to form larger molecules. PLEDs are thin film displays that are created by sandwiching an undoped conjugated polymer between two proper electrodes at a short distance. The polymer emits light when exposed to electricity.
<b>Pre-Generated Fonts</b>	Pre-determined fonts which can be downloaded into graphic liquid crystal displays.
<b>Primitive</b>	A low-level object or operation from which higher-level, more complex objects and operations can be constructed. In graphics, primitives are basic elements, such as lines, curves, and polygons, which you can combine to create more complex graphical images.
<b>RS-232</b>	Short for recommended standard-232C, a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices.
<b>Scroll</b>	To view consecutive lines of data on the display screen. The term scroll means that once the screen is full, each new line appears at the bottom edge of the screen and all other lines move up one position.
<b>Serial Number</b>	A number that is one of a series and is used for identification of the module.
<b>Serial Port</b>	A port, or interface, that can be used for serial communication, in which only 1 bit is transmitted at a time.
<b>Version Number</b>	This refers to the firmware revision number of the module.
<b>Volatile Memory</b>	Temporary memory. Once the power supply is turned off volatile memory is then erased.